# Android应用解析（其它）

## Intent

Android 提供 Intent 机制来协助应用间和应用内部的交互与通信，它是一个数据负载者。当我们需要做一个调用动作，我们就可以通过Intent告诉Android系统来完成这个过程。那么 Andriod 又是怎么知道该让谁来响应这个 Intent 呢？这就需要使用intent-filter 了，可以参考一下捕鱼2的 AndroidManifest.xml

<receiver android:name="org.cocos2dx.UserCenter.AliPayReceiver">

<intent-filter>

<action android:name="android.intent.action.PACKAGE\_ADDED"/>

<data android:scheme="package" />

</intent-filter>

</receiver>

除了上面存在的 action 和 data 属性，intent-filter 还有 category,type,component 等属性。

几个比较常用的 category：

1. Intent.CATEGORY\_DEFAULT（android.intent.category.DEFAULT） 默认的category
2. Intent.CATEGORY\_PREFERENCE（android.intent.category.PREFERENCE） 表示该目标Activity是一个首选项界面；
3. Intent.CATEGORY\_BROWSABLE（android.intent.category.BROWSABLE）指定了此category后，在网页上点击图片或链接时，系统会考虑将此目标Activity列入可选列表，供用户选择以打开图片或链接。

## Content Provider

**内容提供者，比如系统的电话本应用等等，可以把它看成一个网站服务器，而从 Content Provider 中获取数据的应用就是客户端。**

## Service

4.1概念：Android系统的服务，它与activity的主要区别是，service 不能与用户交互且不能自己启动的后台程序，退出应用时service也不会主动结束。

4.2 生命周期：Android Service生命周期与Activity生命周期是相似的，但是也存在一些细节上也存在着重要的不同：

4.2.1 onCreate和onStart是不同的

4.2.2通过从客户端调用Context.startService(Intent)方法我们可以启动一个服务。如果这个服务还没有运行，Android将启动它并且在onCreate方法之后调用它的onStart方法。如果这个服务已经在运行，那么它的onStart方法将被新的Intent再次调用。所以对于单个运行的Service它的onStart方法被反复调用是完全可能的并且是很正常的。

4.2.3 onResume、onPause以及onStop是不需要的

4.3 activity 可以使用bindService()方法和service进行通信。

解疑：已经有Thread了，为什么还要有Service？

Thread 的运行是独立于 Activity 的，也就是说当一个 Activity 被 finish 之后，如果你没有主动停止 Thread 或者 Thread 里的 run 方法没有执行完毕的话，Thread 也会一直执行。因此这里会出现一个问题：当 Activity 被 finish 之后，你不再持有该 Thread 的引用。另一方面，你没有办法在不同的 Activity 中对同一 Thread 进行控制。